ARL2: Aligning Retrievers for Black-box Large Language Models via Self-guided Adaptive Relevance Labeling

Lingxi Zhang[1], **Yue Yu**[2], **Kuan Wang**[2], **Chao Zhang**[2]
[1] Renmin University of China, Beijing, China
[2] Georgia Institute of Technology, Atlanta, USA
zhanglingxi@ruc.edu.cn, {yueyu, kuanwang, chaozhang}@gatech.edu

Abstract

Retrieval-augmented generation enhances large language models (LLMs) by incorporating relevant information from external knowledge sources. This enables LLMs to adapt to specific domains and mitigate hallucinations in knowledge-intensive tasks. However, existing retrievers are often misaligned with LLMs due to their separate training processes and the black-box nature of LLMs. To address this challenge, we propose ARL2, a retriever learning technique that harnesses LLMs as labelers. ARL2 leverages LLMs to annotate and score relevance evidence, enabling learning the retriever from robust LLM supervision. Furthermore, ARL2 uses a adaptive self-training strategy for curating high-quality and diverse relevance data, which can effectively reduce the annotation cost. Extensive experiments demonstrate the effectiveness of ARL2, achieving accuracy improvements of 5.4% on NQ and 4.6% on MMLU compared to the state-of-the-art methods. Additionally, ARL2 exhibits robust transfer learning capabilities and strong zero-shot generalization abilities.

1 Introduction

Retrieval-augmented generation (RAG) is a widely used technique for tailoring large language models (LLMs) (OpenAI, 2023; Anil et al., 2023; Touvron et al., 2023) to specific domains and tasks. By incorporating information from external knowledge sources, RAG enhances LLMs by prompting them with relevant evidence (Lewis et al., 2020; Izacard et al., 2022b; Shi et al., 2023b), without the need for expensive fine-tuning (He et al., 2023). These knowledge sources serve as a non-parametric reference, allowing LLMs to access up-to-date and customized corpora for answering questions. RAG has shown promising results in improving LLM response accuracy for target tasks, while also helping to mitigate LLM hallucination (Ji et al., 2023).

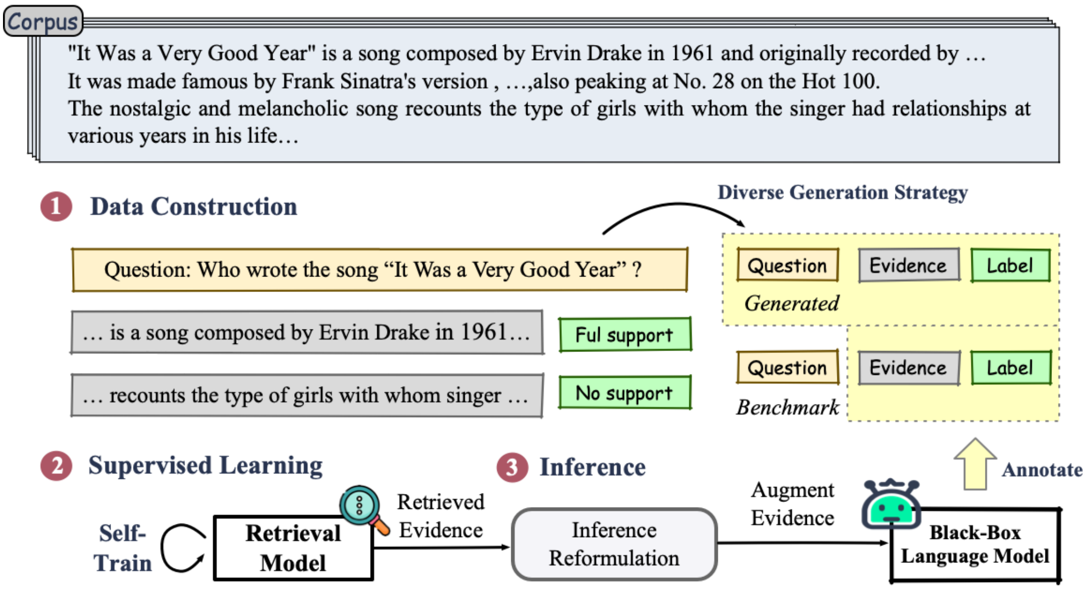
The practice of RAG for state-of-the-art LLMs often involves directly using standard retrievers (e.g. Google Search (Lazaridou et al., 2022), BM25 (Robertson et al., 2009)) or off-the-shelf dense retrievers (e.g., DPR (Karpukhin et al., 2020a), Contriever (Izacard et al., 2022a)) trained with supervised relevance signals. However, the performance of these methods is limited by the mismatch between the retrieval and downstream tasks, as the retrieved *similar* documents may not always be *useful* for the queries despite their relevance. In fact, retrieved documents with similar topics but irrelevant content may even mislead the LLM's predictions (Yu et al., 2023b; Shi et al., 2023a).

To address the challenge of adapting retrievers for LLMs, several works propose joint training of retrievers and language models (Izacard et al., 2022b; Lin et al., 2023b; Cheng et al., 2023). However, these methods require training the LLMs from scratch, which is impractical for cutting-edge LLMs due to their prohibitive training costs and black-box nature. The recent RePlug method (Shi et al., 2023b) offers a solution by refining the retriever for black-box LLMs. RePlug utilizes language modeling scores of the answers as a proxy signal to train the dense retriever. However, such supervision for retriever training is indirect and may not be discriminative enough, especially when the questions could be directly answered through the parametric knowledge of the LLM. Therefore, effectively adapting retrievers for black-box LLMs remains an unsolved challenge.

To enhance the retrieval model's performance, we present our approach, ARL2[1], which leverages guidance from LLMs through self-guided adaptive relevance labeling. Unlike existing methods that rely on indirect supervision via attention or answer-based language modeling scores, ARL2 leverages

[1] Short for **A**ligning **R**etrievers with **L**arge Language Models via Self-guided **A**daptive **R**elevance **L**abeling.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3708–3719
August 11-16, 2024 ©2024 Association for Computational Linguistics



Corpus
"It Was a Very Good Year" is a song composed by Ervin Drake in 1961 and originally recorded by …
It was made famous by Frank Sinatra's version , …,also peaking at No. 28 on the Hot 100.
The nostalgic and melancholic song recounts the type of girls with whom the singer had relationships at various years in his life…
1 Data Construction
Diverse Generation Strategy
Question: Who wrote the song "It Was a Very Good Year" ?
… is a song composed by Ervin Drake in 1961…
Ful support
… recounts the type of girls with whom singer …
No support
Question
Evidence
Label
Generated
Question
Evidence
Label
Benchmark
2 Supervised Learning
3 Inference
Annotate
Self-Train
Retrieval Model
Retrieved Evidence
Inference Reformulation
Augment Evidence
Black-Box Language Model

Figure 1: **Overview of ARL2.** We first construct a diverse and high-quality training set of relevance label through LLM itself (Step 1), then we train the retriever with such relevance supervision (Step 2), finally, we inference the LLM to yield answers through the reformulated augmented documents (Step 3).

the LLMs' capabilities to directly assess document relevance, results in the curation of high-quality relevance labels to train better retrievers.

The advantages of using LLM-annotated relevance labels for retriever training are threefold. Firstly, ARL2 can effectively distinguish truly useful documents from similar but irrelevant ones, providing valuable positives and hard negatives for training. Secondly, it enables the creation of diverse training data beyond a single target dataset, surpassing the limitations of methods like RePlug. Thirdly, ARL2 can reversely generate diverse questions from unlabeled documents, enhancing data diversity and facilitating effective generalization under challenging few-shot or zero-shot scenarios.

To reduce the cost of data curation due to frequent LLM calls for relevance annotation, we propose an adaptive self-training strategy. This strategy empowers the retriever to identify and label confident and well-trained data points, reducing the reliance on costly LLM interactions. Furthermore, we introduce a cluster-driven prompt demonstration metric to ensure the diversity and high quality of the constructed data. This enables learning a strong retriever from a smaller amount of high-quality and diverse annotated data.

Our experiments encompass both open-domain QA datasets (NQ and TQA) and a specific-domain QA dataset (MMLU). The results demonstrate that our framework significantly enhances the performance of both the retriever and RAG-based question answering, achieving an accuracy improvement of 5.4% on NQ and 4.6% on MMLU. Moreover, with the diverse curated data and self-training strategy, our method exhibits strong transferability to specific domains with limited training data and delivers promising results even in the challenging zero-shot generalization setting.

In summary, our contributions are as follows: (1) We propose ARL2, a retrieval augmentation framework that effectively aligns retrieval models with black-box LLMs. ARL2 leverages the LLM as a labeler to assign relevance scores with robust LLM self-guided supervision. (2) We incorporate a cluster-driven prompt demonstration metric to ensure the generation of high-quality data. Additionally, we explore a self-training strategy for the retriever to reduce the computational cost of LLM calls. (3) Extensive experiments demonstrate that our retrieval augmentation framework not only improves the performance of LLMs across various question-answering tasks but also exhibits strong transfer and zero-shot generalization capabilities.

2 Related Work

Dense Retrieval. Earlier research has explored various ways to learn representations for text retrieval (Deerwester et al., 1990; Huang et al., 2013; Gillick et al., 2018). With the rise of pre-trained language models, several works have presented the

BERT-based dual-encoder as dense retrievers (Lee et al., 2019; Karpukhin et al., 2020a; Xiong et al., 2021). They typically employ encoders to independently encode queries and documents into a dense space, calculating the similarity via vector dot-product or cosine similarity. To further enhance performance of dense retrieval models, one line of approaches focuses on developing retrieval-oriented pretraining techniques (Izacard et al., 2022b; Gao and Callan, 2021, 2022; Yu et al., 2022; Xiao et al., 2022; Lin et al., 2023a), and another line of approaches focus on improving the negative contrast loss (Ren et al., 2021; Zhang et al., 2022). Additionally, some models utilize LLM-generated queries to generate synthetic examples for improving retrieval (Ma et al., 2021; Dai et al., 2023), but these dense retrievers are trained separately from LLM and may not always align well with the LLM, potentially resulting in sub-optimal performance when directly applied to target tasks (Lin et al., 2023b).

Retrieval-Augmented LLMs. RAG have been widely used for language modeling (Borgeaud et al., 2022; Ram et al., 2023), question answering (Lewis et al., 2020; Izacard et al., 2022b; Shi et al., 2023b), and domain adaptation (Xu et al., 2023a,b; Shi et al., 2023c). To align retrievers with LLMs, most RAG methods integrate a pre-trained retriever with a generator and subsequently undergo an end-to-end fine-tuning process to effectively capture knowledge (Lewis et al., 2020). Among them, Atlas (Izacard et al., 2022b) leverages retrieved documents as latent variables and fine-tunes retrieval models with four designed loss. AAR (Yu et al., 2023c) identifies the LLM's preferred documents through FiD cross-attention scores (Izacard and Grave, 2021), and fine-tuning the retriever with hard negative sampling. However, these methods are inapplicable to black-box LLM as they require accessing LLM parameters. The only exception is RePlug (Shi et al., 2023b), which conducts supervised training by evaluating the KL divergence between the probability distributions of the retrieved documents and LLM's likelihood.

3 Methodology

In ARL2, we employ LLMs to explicitly label relevance scores between questions and evidence, thereby generating relevance supervision for training an LLM-aligned retriever. ARL2 addresses two key challenges: (1) How can we effectively utilize the LLM to construct a diverse and high-quality training set of relevance labels? (Section 3.1) and (2) How can we train the retriever using the provided relevance supervision and further leverage the retriever to inform adaptive LLM annotation? An overview of our proposed ARL2 method is presented in Figure 1.

3.1 Data Construction

To collect labeled data for retriever learning, it is crucial to go beyond query-document relevance as in standard information retrieval (Lee et al., 2019). In fact, the question q in RAG applications is often under-specified and requires deeper language understanding. Motivated by this, we leverage LLMs to provide direct supervision signals on the *usefulness* for each piece of evidence for the question.

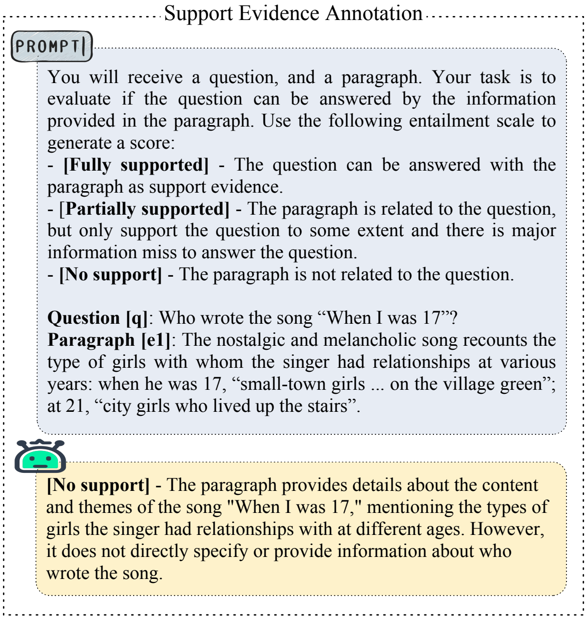
Specifically, we create training tuples denoted as $\mathcal{T} = (q_i, d_i, e_i, s_i)_{i=1}^{|\mathcal{T}|}$, where the evidence e is a text segment extracted from the document d, and the variable s represents the level of support for the question q based on the evidence e. The relevance score s can take on three values: 0 for "*no support*", 0.5 for "*partial support*", and 1 for "*full support*".

We construct the corpus $\mathcal{D}$ by compiling various corpora, such as WikiPedia (Vrandečić and Krötzsch, 2014) and MS MARCO (Bajaj et al., 2016). For each document d in $\mathcal{D}$, we generate a training tuple through a three-step process: *question generation*, *evidence identification*, and *evidence scoring*. This procedure yields a dataset of 100,000 annotated instances, denoted as $\mathcal{T}_g$. Additionally, we curate 140,000 data instances $\mathcal{T}_b$ from QA benchmarks. The combination of these two sets forms the complete dataset $\mathcal{T}$.

3.1.1 Question Generation

To avoid manually generating questions from a vast amount of documents, we employ ChatGPT (`gpt-3.5-turbo-0613`) for question generation to generate pairs in the format of (q, e) from documents in the corpus $d \in \mathcal{D}$. This process involves providing the LLM reader with a specific prompt, namely, "*Given only the information below, following the examples, ask a factual question that we can answer according to the given passage*", along with demonstrations to steer in-context learning.

Diverse Generation Strategy. The quality of generated questions heavily relies on the selected document and the provided question examples. Often, the generated questions follow similar patterns as the examples given. Here, the question pattern refers to the sentence structure of the questions,


Support Evidence Annotation

You will receive a question, and a paragraph. Your task is to evaluate if the question can be answered by the information provided in the paragraph. Use the following entailment scale to generate a score:
- **[Fully supported]** - The question can be answered with the paragraph as support evidence.
- **[Partially supported]** - The paragraph is related to the question, but only support the question to some extent and there is major information miss to answer the question.
- **[No support]** - The paragraph is not related to the question.

Question [q]: Who wrote the song “When I was 17”?
Paragraph [e1]: The nostalgic and melancholic song recounts the type of girls with whom the singer had relationships at various years: when he was 17, “small-town girls ... on the village green”; at 21, “city girls who lived up the stairs”.

[No support] - The paragraph provides details about the content and themes of the song "When I was 17," mentioning the types of girls the singer had relationships with at different ages. However, it does not directly specify or provide information about who wrote the song.

Figure 2: Illustration of prompting LLM to annotate relevance labeling data.

such as special questions, yes-no questions, and declarative sentences. To enhance the quality of the constructed data, we employ a diverse selection strategy. First, we select relevant questions q_r from $\mathcal{T}_b$ that share the same or a similar domain with the target document. Second, we cluster the questions in $\mathcal{T}_b$ based on their patterns and opt for multiple question patterns from different clusters to generate diverse questions. To facilitate diverse and high-quality question generation, we mask the entity mentions in the questions to focus on their core structure. Each masked question is then mapped into a sentence embedding and clustered using ISODATA (Ball et al., 1965), a variation of K-means. This makes the generated questions not only diverse but also of high quality.

3.1.2 Support Evidence Annotation

Evidence Identification. For each data point (q_i, d_i, e_i) in $\mathcal{T}_b$, we assign a support level label s to indicate the relevance of the evidence e_i to the question q_i. If (q_i, d_i, e_i) is directly obtained from human annotators, we set s to *“full support”*. Otherwise, we use ChatGPT as an annotator to obtain the support level label. Specifically, we provide ChatGPT with the question q_i, the document d_i, and an instruction to extract the supporting evidence from the document. ChatGPT then returns the extracted evidence and its relevance score to the question.

Evidence Scoring. In addition to the positive samples with *“full support”* labels, we also create negative samples for training the retrieval model. For each positive sample (q_i, d_i, e_i), we construct a challenging negative set $\mathcal{N}_i$ by selecting evidence segments from the same document d_i or from other documents that are content- or domain-similar to d_i. We embed each evidence segment in $\mathcal{N}_i$ using SimCSE (Gao et al., 2021), a sentence embedding model. Then, we identify the top-k most semantically similar evidence segments to e_i based on their cosine similarity scores. Finally, we ask ChatGPT to label the relevance of these top-k evidence segments to the question q_i. We discard the evidence segments labeled as *“full support”* and retain the ones labeled as *“partial support”* or *“no support”* as negative samples.

3.2 Retrieval Model Learning

In ARL2, we train a dense retriever from the above instances of <question, evidence, relevance score>. To enhance the learning of the retriever, we design pairwise and list-wise losses that incorporate hard negative sampling. Furthermore, to reduce the cost associated with ChatGPT annotation, we propose a self-training strategy that enhances efficiency.

3.2.1 Dense Retriever

Given a query q and an evidence corpus $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$, the goal of the dense retriever is to first map all the documents $d \in \mathcal{D}$ in a dense vector to build an index for retrieval, then retrieve the top-k most relevant documents via efficient vector similarity metrics.

We leverage the dual-encoder structure (Lee et al., 2019; Karpukhin et al., 2020b; Lin et al., 2023a) to embed query and documents using text encoders initialized from pretrained language models. Specifically, the query encoder $E_Q(\cdot)$ and document encoder $E_D(\cdot)$ map queries and documents $d \in \mathcal{D}$ to low-dimensional real-valued vectors:

$$\mathrm{sim}(q, d) = \cos\left(E_Q(q), E_D(d)\right). \qquad (1)$$

For efficient retrieval, we pre-compute the embedding of each document in $\mathcal{D}$ and construct a FAISS index (Johnson et al., 2019).

3.2.2 Learning Objective for Dense Retreivers

To train the retriever, we employ a ranking loss. For each query q_i, we obtain the positive candidate e^+ along with a list of negative evidence candidates $\mathcal{N}_i = \{e_1, e_2, ...\}$. For each negative sample $e_j \in \mathcal{N}_i$, we obtain its corresponding predicted relevance score $\widehat{s}_{ij} = \mathrm{sim}(q_i, e_j)$ with the dual encoder and labeled relevance score s_{ij} via LLM prompting (Figure 2). In specific, we optimize the retriever by

$$\mathcal{L} = \mathcal{L}_{\text{list}} + \mathcal{L}_{\text{pair}},$$

where $\mathcal{L}_{\text{list}}$ indicates a list-wise loss that discriminates between the positive and all listed negative evidence, and $\mathcal{L}_{\text{pair}}$ indicates a pairwise loss which focus on pairwise comparison between both ("*full support*" ,"*partial support*") and ("*partial support*","*not support*").

List-wise Contrastive Loss. We utilize an InfoNCE loss which encourages positive instances to have high scores and negative instances to have low scores, which is defined as:

$$\mathcal{L}_{\text{list}}\left(s_i, \widehat{s}_i\right) = -\frac{\exp\left(\widehat{s}_i^{+}/\tau\right)}{\exp\left(\widehat{s}_i^{+}/\tau\right) + \sum_{j=1}^{|\mathcal{N}_i|}\exp\left(\widehat{s}_{ij}/\tau\right)}$$

where $\widehat{s}_i^{+}$ is the score for positive instance which is labeled as "*full support*", and τ is a temperature parameter.

Pairwise Logistic Loss. To better capturing the fine-grained relevance information beyond binary relevance labels, we further leverage a pairwise loss to deal with the "*partial support*", which is defined as follow,

$$\mathcal{L}_{\text{pair}}\left(s_i, \widehat{s}_i\right) = \sum_{j=1}^{|\mathcal{N}_i|}\sum_{j'=1}^{|\mathcal{N}_i|}\mathbb{I}_{s_{ij}>s_{ij'}}\log\left(1 + e^{\widehat{s}_{ij'} - \widehat{s}_{ij}}\right)$$

The aim of employing pairwise logistic loss is to ensure that a partially supported document can achieve a higher score than a fully negative one, while still scoring lower than a fully supported positive document.

Negative Sampling. Another important aspect of the above learning objective is how to mine negative examples $\mathcal{N}_i$ for each query q_i. Here we propose a multi-step bootstrapped strategy to gradually provide "*harder*" negative examples to effectively train the retriever. Initially, we kickstart the retriever training using in-batch negative samples as the warmup (Gillick et al., 2018; Lee et al., 2019). Then, we choose the "*partial support*" document as hard negative for both loss for further model training. This advanced negative sampling strategy significantly improves the retriever's performance compared to using randomly selected or BM25 selected negative samples (Karpukhin et al., 2020b).

3.2.3 Adaptive Relevance Labeling

Although data collection can proceed without human annotations, utilizing ChatGPT series APIs comes with associated costs. To mitigate these expenses, we propose an adaptive strategy wherein we initially label only a subset of support evidence training data $\mathcal{T}_s$ first. After the warm-up epochs trained by $\mathcal{T}_s$, given a (q_i,d) pair without annotation, we segment document d into text chunks $d = \{e_{i,1}, ..., e_{i,n}\}$ and enable the retriever to select the support evidence e_{ij} with highest relevance score as

$$s(q_i) = \max_{j\in\{1,2,\dots,n\}} \mathrm{sim}(q_i, e_{i,j}).$$

Such a confidence score $s(q_i)$ reflects the model's confidence on question q_i. Rather than evaluating confidence for each question individually, we first cluster questions (following the same pipeline as in Sec. 3.1.1) and calculate the confidence score for each cluster $C_1, C_2, ..C_m$, which denotes as

$$s(C_i) = \sum_{q_t\in C_i} s(q_t)/|C_i|.$$

When a cluster exhibits a high model confidence score, we depend on the retriever's predictions and integrate all confident predictions within the cluster as supplementary training data. Specifically, we fine-tune the retriever using both the original training data and the newly generated training data in the subsequent epoch. Conversely, if a cluster's confidence score is low, we resort to the data construction process outlined in Sec. 3.1, which utilizes ChatGPT to create more relevance labels.

3.3 Inference

Formally, the LLM conditions on both the input question q and the support evidence $\mathcal{D}_q$ to generate a textual output y as the answer. To augment the support evidence, a simple way is to input the evidence in D_q from the highest relevant to the least relevant. However, the LLM is shown to be not robustly accessing or using information in long input contexts and may lose information in the middle (Liu et al., 2023). Therefore, we reorder the top-k documents, placing the most relevant document at either the beginning or the end of its input context. Concretely, we cut the D_q into three subsets according to the relevance score and put d_1 to d_j at the beginning, d_{j+1} to d_{2j} at the end, and the rest at the middle, showing as following

$$R_1 = d_1 \circ ... \circ d_j \circ d_{2j+1} \circ ... \circ d_k \circ d_{2j} \circ ... \circ d_{j+1}$$

where the $\circ$ denotes the concatenation of two sequences. To ensure robustness, we further rearrange the order within each subset for N times to obtain permutations $\mathcal{R} = \{R_1, \ldots, R_n\}$, and then ensemble the likelihood provided by the LLM for each R_i to calculate the final answer score as

$$p(y|q, D_q) = \sum_{i=1}^{N} P(y|q \circ R_i).$$

4 Experiments

4.1 Experimental Setting

Datasets. We evaluate the effectiveness of our framework through various open domain QA datasets, encompassing both general tasks including Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017) as well as domain-specific tasks like MMLU (Hendrycks et al., 2021). NQ comprises natural questions extracted from real Google search, while TQA emphasizes trivia questions. MMLU is a multiple-choice QA benchmark which can be categorized into humanities, STEM, social sciences, and others. For the document corpus, we follow DPR (Karpukhin et al., 2020a), and construct a corpus of 300,000 QA pairs in total, which consists of 8.8 million passages from MS MARCO (Bajaj et al., 2016) and about 21 million passages from Wiki dump (Vrandečić and Krötzsch, 2014). The size of each dataset is detailed in Table 1.

Baselines. We compare our methods with strong baselines for open-domain question-answering tasks, which can be divided into: traditional joint training approaches and LLM-based approaches. Specifically, traditional joint training approaches encompass RETRO (Borgeaud et al., 2022), UnifiedQA (Khashabi et al., 2020), and Atlas (Izacard et al., 2022b). The LLM-based approach involves prompting the LLM to generate answers to questions. Representative LLM examples in our evaluation include Chinchilla (Hoffmann et al., 2022), PaLM (Anil et al., 2023), ChatGPT (OpenAI, 2023), Codex (Chen et al., 2021), GenRead (Yu et al., 2023a). To establish stronger baselines, besides raw LLMs, we compare our methods with retrieval augment methods which incorporating well-trained retrievers such as Contriever (Izacard et al., 2022a), Replug (Shi et al., 2023b), both in a few-shot training setting, and also we evaluate the fully supervised DPR (Karpukhin et al., 2020b).

Evaluation Metrics. We evaluate the Recall@Top-k (k = 10, 20) for retriever performance, measuring whether the answer match a text span within the Top-k retrieved passages. For the overall precision, we consider accuracy (Acc.), verifying whether the answer generated by the RALM exactly matches the ground true answer in the dataset.

Table 1: Number of questions in each QA dataset.

Dataset	Train	Dev	Test	Task
NQ	79,168	8,757	3,610	Open-Domain
TriviaQA	78,785	8,837	11,313	Open-Domain
MMLU	15,908	1,540	14,079	Multi-Choice
MS MARCO	1,010,916	–	–	Retrieval

Settings. Based on DPR, we trained ARL2 with generated training instances based on the above corpus along with the benchmark dataset. Additionally, we evaluate different versions of ARL2. **ARL2 (few-shot)** is a few-shot variant which trained on contriever and solely on generated instances along with few-shot benchmark training data. Furthermore, we examine **ARL2 (w/ re-ranker)**, an enhanced version that utilizes a generation-based reranker to improve retriever performance. Specifically, we employ the retriever to fetch the top 50 passages for each question and segment each document into text chunks for re-ranking, which ensuring to be within 50 words or 3 sentences. The re-ranker is trained with the same loss and training instances as the retriever. We evaluate all versions of our framework on ChatGPT with 20-shots.

4.2 Overall Result

The overall results are shown in Table 2,with bold indicating the best and underline indicating the second-best performances.

Our retriever can enhance LLMs: Both "ARL2" and "ARL2 *(w/ re-ranker)*" outperform raw ChatGPT demonstrating that the retrieved evidence can correct the factual errors within ChatGPT's in-parameter knowledge. Especially, we exhibit a remarkable 6% gain on MMLU, as MMLU involves extensive knowledge in a specific domain and certain numerical statistics that are often messed in ChatGPT's in-parameters. Our retrieved evidence contains the correct statistics, which can assist ChatGPT in correcting its answers.

ARL2 surpasses other retrieval augmentation methods, including strong baseline RePlug and other traditional retrievers. These outcomes suggest that our model adapts more effectively to LLM because it's trained on LLM-labeled data, ensuring that the retrieved evidence aligns better with

Table 2: Overall Accuracy on QA datasets (%).

	Natural Question	**TriviaQA**	**MMLU**				
Model	All	All	All	Hum.	Soc.	STEM	Other
Chinchilla (Hoffmann et al., 2022)	35.3	64.7	67.5	63.6	79.3	55.0	73.9
PaLM (Anil et al., 2023)	39.6	–	69.3	77.0	81.0	55.6	69.6
Codex (Chen et al., 2021)	40.6	73.6	68.3	74.2	76.9	57.8	70.1
ChatGPT (OpenAI, 2023)	38.7	74.2	70.3	75.1	76.8	59.3	70.8
GenRead (Yu et al., 2023a)	54.0	74.3	–	–	–	–	–
RETRO (Borgeaud et al., 2022)	45.5	69.9	–	–	–	–	–
UnifiedQA (Khashabi et al., 2020)	55.9	–	48.9	45.6	56.6	40.2	54.6
Atlas (Izacard et al., 2022b)	60.4	79.8	66.0	61.1	77.2	53.2	74.4
ChatGPT+Contriever (Izacard et al., 2022a)	44.2	76.0	69.9	68.1	76.6	50.8	74.8
ChatGPT+DPR (Karpukhin et al., 2020b)	58.0	76.9	72.9	69.6	80.6	64.2	78.6
ChatGPT+Replug (Shi et al., 2023b)	45.4	77.8	71.8	76.5	79.9	58.9	73.2
ChatGPT+ARL2	62.3	82.4	75.7	73.2	80.9	65.5	80.1
ChatGPT+ARL2 (few-shot)	54.9	81.0	73.9	70.8	80.4	66.7	78.8
ChatGPT+ARL2 (w/ re-ranker)	**65.9**	**85.9**	**76.4**	**78.3**	**83.2**	**68.0**	**82.7**

Table 3: Performance on few/zero-shot transfers (%).

	MMLU (Few-shot)			TQA (Zero-shot)	
	Soc.	Hum.	STEM	All	R@20
PaLM	81.0	77.0	55.6	–	–
Atlas	54.6	46.1	52.8	79.8	–
ChatGPT	75.8	73.4	69.9	74.2	–
ChatGPT+Replug	79.9	76.0	72.1	77.8	76.2
ChatGPT+Contriver	80.3	74.1	71.9	76.0	74.2
ChatGPT+ARL2	**81.7**	**79.2**	**74.4**	**77.9**	**77.2**

LLM requirements. Additionally, we compare our model with some joint training baselines (RETRO, Atlas and UnifiedQA), owing to the strong capabilities of ChatGPT, augmenting black-box LLM with retrieved evidence effectively reduces the issue of hallucinations and surpasses joint training of medium-size PLM.

Our re-ranker significantly improves retrieval performance, as indicated in the table. "ARL2 *(w/ re-ranker)*" outperforms pure retriever baselines. The re-ranker divides passages into multiple pieces of evidence, enabling a fine-grained ranking with our adaptive labeling strategy. Because the evidence is shorter than the entire passage and exhibits higher recall with a stronger base model, the re-ranker provides a more concise and accurate augmentation for the language model. This is particularly advantageous due to input length limitations imposed by the language model, resulting in a much better performance compared to "ARL2".

4.3 Generalization and Transfer Ability

Few-shot Transfer Performance. We first evaluate the few-shot transfer ability of our retriever. In our training process, we solely pre-trained the retriever using data derived from or constructed from the MS MARCO datasets which only contain document corpus and questions, following Contriever. Subsequently, we employed few-shot questions (setting as 20 in our experiment) from MMLU as in-context learning examples to generate additional training data exclusively based on the document corpus. As the result shown in the left two columns of Table 3, our model has surpassed both the raw LLM and other baseline retrieval methods, indicating its superior transfer capabilities. As our model fully utilizes the few-shot examples, generating training data patterns similar to those in the MMLU dataset, our retriever is familiar with the setting in MMLU even with a few data.

Zero-shot Generalization Performance. We evaluate the zero-shot ability by training a retriever solely on the benchmark data in the NQ dataset and the constructed data from the NQ corpus, then directly test the model on TQA dataset. As shown in right column of Table 3, the result demonstrates that our retriever has stronger generalization abilities. Additionally, we provide the retriever's recall rate to emphasize that our model's transfer ability isn't solely attributable to ChatGPT's robust generalization capabilities, the diverse select strategy has imparted a level of generalization to the retriever itself by maintaining a diverse range of training data across various domains and patterns.

4.4 Ablation Study

As shown in Table 4, we here perform the ablation study to demonstrate the impact of each strategy in construct relevance label and training the retriever.

Table 4: Ablation study on NQ and MMLU. (%)

Methods	NQ		MMLU
	Acc	R@20	Acc
ARL2	62.3	84.3	75.7
Effect of Learning Objective			
w/o pairwise	59.0	82.1	75.2
w/o listwise	58.5	81.2	74.8
w/o neg sample	62.1	77.0	73.3
Effect of Relevance Labeling			
w/o partial	55.4	77.3	69.5
w/o label data	48.7	68.5	68.2
Effect of Inference Reformulation			
w/o ensemble	59.7	–	75.6
w/o middle-rank	59.1	–	75.2

Impact of Ranking Loss. Our final model leverage both pairwise and list-wise, and we here perform three models with different loss setting, *"w/o pairwise"* indicates training the retriever without the pairwise loss, *"w/o listwise"* indicates training the retriever without the list-wise loss. The results show that remove pairwise loss and list-wise loss will drop recall, and leveraging both loss can achieve a better performance, as it both consider multiple random simple negative evidence to warm up but also take consider hard negative evidence through the pairwise to further improve models' performance. We also evaluate *"w/ neg sample"* which replace our negative sample strategy with BM25 negatives (Karpukhin et al., 2020b), and the results show the efficient of our strategy by bring harder cases for model training.

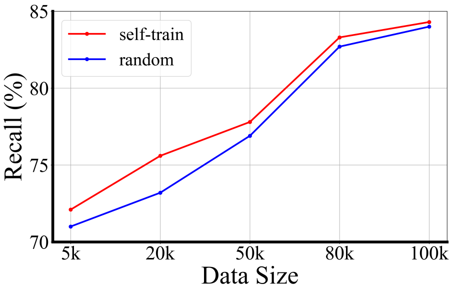


Figure 3: Effect of annotated data size on NQ.

Impact of Relevance Label Training Instance. Our labeled data can contribute the retriever's performance mainly on two aspect, the labeling of partial support evidence and the construction of more training data. So we perform ablation study on these two aspect, where *"w/o partial"* denotes removing all partial label, and *"w/o label data"* denotes removing all annotated data. The results show that both these removing drop accuracy, but *"w/o label data"* drop rapidly, which indicates that more data is more important than detailed training data.

Impact of Less Training Data. Considering the costly nature of API calls for data generation, we evaluate model performance with a smaller subset as training data. As shown in Figure 3, the result illustrates that increased training data leads to enhanced performance, with optimal results achieved when all generated data are utilized. However, the rate of improvement declines with the increasing of training data, indicating that by selectively labeling diverse data, costs can be reduced while maintaining comparable performance. Compare to the diverse self-train strategy, we also show the performance of randomly selection and training solely based on it. As shown, the self-train method can save the cost and achieve better result within same less data, which is more efficient.

Impact of Inference Reformulation. For inference, analysis the impact of two input reformulation strategies, *"w/o ensemble"* indicates the model directly takes the result from middle-rank input, *"w/o middle-rank"* indicates the model take a simple positive rank input. And the result shows that both these two strategy can benefit LLM's performance. The accuracy does not drop much indicates that although model may lost information in middle, LLM can still obtain some external factual information from the input augment data.

5 Conclusion

We introduce ARL2, a retrieval-augmentation framework that harnesses the capabilities of LLMs as annotators for retriever learning. Unlike conventional approaches that face misalignment due to separate training processes and the inherent complexity of LLMs, our framework dynamically leverages LLMs to annotate and assess relevant evidence. This enables the retriever to benefit from robust LLM supervision. Additionally, we incorporate a self-training strategy to mitigate the cost associated with API calls. Through extensive experimentation, we demonstrate the effectiveness of ARL2, which enhances accuracy in open-domain QA tasks, exhibits robust transfer learning capabilities, and showcases strong zero-shot generalization abilities.

Acknowledgements

We would like to thank reviewers from the ACL Rolling Review for the helpful feedback. This work was supported in part by NSF IIS-2008334 and CAREER IIS-2144338.

Limitations

In this research, annotating relevance labels can be expensive due to the extensive use of ChatGPT APIs. Our future work will further explore strategies to generate diverse, high-quality data to reduce these costs. Additionally, we aim to expand the curated relevance data to cover more specific domains like biomedical and life sciences. Correspondingly, we will evaluate the method's performance on tasks from such domains to assess its generalizability.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Geoffrey H Ball, David J Hall, et al. 1965. *ISODATA, a novel method of data analysis and pattern classification*, volume 4. Stanford research institute Menlo Park, CA.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, pages 2206–2240. PMLR.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self-memory. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*.

Tao He, Xue Li, Zhibin Wang, Kun Qian, Jingbo Xu, Wenyuan Yu, and Jingren Zhou. 2023. Unicron: Economizing self-healing llm training at scale. *arXiv preprint arXiv:2401.00134*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020b. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Weixian Waylon Li, Yftah Ziser, Maximin Coavoux, and Shay B. Cohen. 2023. BERT is not the count: Learning to match mathematical statements with proofs. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3581–3593, Dubrovnik, Croatia. Association for Computational Linguistics.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023a. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore. Association for Computational Linguistics.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023b.

Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1075–1088, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Wenqi Shi, Yuchen Zhuang, Yuanda Zhu, Henry Iwinski, Michael Wattenbarger, and May Dongmei Wang. 2023c. Retrieval-augmented large language models for adolescent idiopathic scoliosis patients in shared decision-making. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–10.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-training retrieval-oriented language models via masked auto-encoder. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 538–548, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Benfeng Xu, Chunxu Zhao, Wenbin Jiang, PengFei Zhu, Songtai Dai, Chao Pang, Zhuo Sun, Shuohuan Wang, and Yu Sun. 2023a. Retrieval-augmented domain adaptation of language models. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 54–64, Toronto, Canada. Association for Computational Linguistics.

Ran Xu, Yue Yu, Joyce Ho, and Carl Yang. 2023b. Weakly-supervised scientific document classification via retrieval-augmented multi-stage training. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2501–2505.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023a. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023b. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. COCO-DR: Combating distribution shift in zero-shot dense retrieval with contrastive and distributionally robust learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023c. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial retriever-ranker for dense text retrieval. In *International Conference on Learning Representations*.

A Appendix

A.1 Implementation Details

To enhance the efficiency of the training process, we establish a FAISS index for rapid similarity searches. To train the model, we employ the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 2e-5, a batch size of 256, and a warm-up ratio of 0.1 during training. Following (Shi et al., 2023b), document embeddings are refreshed every 5k steps. In the case of few-shot learning, we adhere to the setup in contriever (Izacard et al., 2022a), utilizing a momentum value of 0.9995 and a temperature of 0.05, and a learning rate of 5e-5, a batch size of 512. The dense retriever is initialized with `BERT-base-uncased` model (Devlin et al., 2019).

A.2 Diverse Generation Details

We leverage BERT (Li et al., 2023) to embed the questions in the pool. In order to ensure the diversity of demonstrations, we employ the classic K-Means algorithm to cluster questions based on their embeddings. The number of clusters is set to 6, and the structure of questions can be YesNo, What/When, Which, Numeric, Location, Person. We select questions by randomly choosing one question from each group. Notably, we mask entity mentions in each question to allow the clustering algorithm concentrate on the sentence structure information rather than specific instances.